Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

Listing of Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

1   1.   (Previously presented): A method for classifying received network data
2   comprising:
3   providing a language definition; and
4   processing incoming network data with said language definition in accordance
5   with a formal language processing technique including scanning said network data using lexical
6   token scanning according to said language definition, wherein said network data is treated as a
7   stream of input bytes, said network data being organized into data packets, said scanning
8   resulting in the identification of a data packet as belonging to one of a plurality of classes.

1   2.   (Original): The method of claim 1 wherein said scanning includes
2   identifying an arithmetic operation and performing said arithmetic operation.

1   3.   (Original): The method of claim 1 wherein said scanning includes
2   identifying a skip operation and in response thereto skipping over one or more subsequent input
3   bytes.

1   4.   (Original): The method of claim 1 wherein said lexical scanning includes
2   providing a set of regular expressions, each regular expression having an associated class
3   identifier.

1   5.   (Original): The method of claim 1 further including providing a
2   deterministic finite automaton (DFA) comprising plural states, said step of scanning including
3   recognizing data packets using said DFA including transitioning from one state to another.

1   6.   (Original): The method of claim 5 wherein said data packets are variable
2   length data packets.

Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

1   7.   (Original): The method of claim 5 wherein said DFA is defined by a set

2 of regular expressions.

1   8.   (Original): The method of claim 7 further including generating a grammar

2 tree data structure representative of said regular expression, producing a non-deterministic finite

3 automaton (NFA) from said grammar tree data structure, and converting said NFA to produce

4 said DFA.

1   9.   (Original): The method of claim 5 wherein some of said states include

2 one or more associated computer instructions and wherein said computer instructions are

3 executed in connection with transitioning to a state.

1   10.   (Original): The method of claim 9 wherein some of said states further

2 include a skip instruction.

1   11.   (Previously presented): In a network data switching device, a method for

2 classifying data packets comprising steps of:

3   providing a language definition in the form of one or more regular expressions,

4 each having an associated class identifier;

5   receiving plural data packets, each having a length not necessarily equal to one

6 another; and

7   for each data packet, processing said data packet in accordance with a formal

8 language processing technique including determining a matching regular expression from among

9 said regular expressions that matches said data packet, wherein said each data packet is classified

10 according to the class identifier associated with said matching regular expression.

1   12.   (Original): The method of claim 11 wherein said data packets comprise a

2 data stream and said determining includes lexically scanning said data stream.

1   13.   (Original): The method of claim 11 wherein said regular expressions are

2 represented by a deterministic finite automaton (DFA).

Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

1   14.   (Original): The method of claim 13 wherein said DFA is in compressed

2   form.

1   15.   (Original): The method of claim 11 further including compiling said

2   regular expressions to produce said DFA.

1   16.   (Original): The method of claim 15 wherein said compiling produces a

2   non-deterministic finite automaton (NFA) as intermediate data structure, said compiling further

3   includes converting said NFA to produce said DFA.

1   17.   (Original): The method of claim 16 further including reducing said DFA

2   to a compressed form.

1   18.   (Original): The method of claim 11 wherein said data packet comprises

2   plural bytes, and said determining includes detecting an operator indicating a number of bytes to

3   be skipped.

1   19.   (Original): The method of claim 18 wherein said number is specified by

2   the value of a current input byte.

1   20.   (Original): The method of claim 18 wherein said number is specified in a

2   register.

1   21.   (Original): The method of claim 18 wherein said determining further

2   includes detecting an operator indicating a value to be saved in a register.

1   22.   (Original): The method of claim 21 wherein said determining further

2   includes detecting an operator indicating a logical or mathematical operation to be performed on

3   the contents of said register.

Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

1   23.   (Previously presented): In a data packet receiving and forwarding device,

2   a method for classifying received data packets comprising a stream of data, said method

3   comprising steps of:

4   receiving a description of classification rules in the form of a classification

5   language definition;

6   compiling said classification language definition to produce a deterministic finite

7   automaton (DFA) comprising plural states;

8   configuring a programmable hardware packet classifier with said DFA; and

9   receiving said data stream and processing said data stream in accordance with a

10   formal language processing technique including scanning said data stream with said hardware

11   packet classifier to classify said received data packets.

1   24.   (Original): The method of claim 23 wherein said compiling includes

2   associating arithmetic and logic instructions with some of said states.

1   25.   (Original): The method of claim 23 wherein said classification language

2   includes regular expressions.

1   26.   (Original): The method of claim 25 wherein said regular expressions

2   include arithmetic and logic operations.

1   27.   (Original): The method of claim 26 wherein said regular expressions

2   further include skip operations.

1   28.   (Original): The method of claim 27 wherein said regular expressions

2   further include data storage operations.

1   29.   (Original): The method of claim 23 wherein said DFA is in compressed

2   format.

1   30.   (Previously presented): The method of claim 23 further including:

2   receiving a second description of classification rules in the form of a second

3   classification language definition;

4   compiling said second classification language definition to produce a second

5   DFA;

6   configuring a programmable hardware packet classifier with said second DFA;

7   and

8   applying said data stream to said hardware packet classifier to classify said

9   received data packets,

10   wherein said data packets are classified according to said second classification

11   rules, thereby facilitating changing packetizing policies in said data packet routing device.

1   31.   (Previously presented): A network data packet classifier comprising:

2   an input port for receiving network data packets comprising a stream of data;

3   a memory assemblage configured with data representing a deterministic finite

4   automaton (DFA), said DFA representing plural regular expressions according to a language

5   definition; and

6   decompression logic operatively coupled to said memory assemblage and

7   configured to process said stream of data using said language definition in accordance with a

8   formal language processing technique including scanning said stream of data with said DFA to

9   find a matching one of said regular expressions,

10   said regular expressions having corresponding class identifiers,

11   wherein each of said network data packets is associated with the class identifier of

12   said regular expression that matches said each network data packet.

1   32.   (Original): The classifier of claim 31 wherein some of said regular

2   expressions include arithmetic instructions and logic instructions, said memory assemblage

3   further configured to contain said instructions, the classifier further including an arithmetic logic

4   unit operatively coupled to said decompression logic and configured to execute said instructions.

1   33.   (Original): The classifier of claim 32 further including at least one register

2   operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to

3   store data into said register in response to a save instruction.

1   34.   (Original): The classifier of claim 32 further including skip logic

2   operatively coupled to said logic component and configured to skip over an amount of data in

3   response a skip instruction.

1   35.   (Original): The classifier of claim 31 wherein said network data packets

2   can vary from one packet to another.

1   36.   (Original): The classifier of claim 31 wherein said DFA is in compressed

2   form.

1   37.   (Original): The classifier of claim 36 wherein said DFA comprises plural

2   non-default states and plural default states, and said memory assemblage comprises a base

3   memory, a next-state memory, and a default-state memory; said base memory configured to

4   contain address locations of said next-state memory, said next-state memory representing all of

5   said non-default states, said default-state memory representing all of said default states.

1   38.   (Original): The classifier of claim 37 wherein said memories are random

2   access memories.

1   39.   (Original): The classifier of claim 37 wherein said memories are read-

2   only memories.

1   40.   (Previously presented): A network data packet classifier comprising:

2   an input configured to provide a data packet comprising a stream of data;

3   a first system of memory configured with data representing a deterministic finite

4   automaton (DFA), said DFA defined in accordance with a language definition and comprising

5   plural states including an initial state and plural terminating states;

Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

6   a system of logic circuits operatively coupled to said first system of memory and

7   to said input, and configured to process said data stream using said language definition in

8   accordance with a formal language processing technique including a step to lexically scan said

9   data stream with said DFA to produce a reached terminating state; and

10   a second system of memory configured with data representing a class index

11   corresponding to each of said terminating states and configured to output a class index in

12   response to the production of said reached terminating state.

1   41.   (Original): The classifier of claim 40 further including a third system of

2   memory configured to contain current state information for plural input channels, said system of

3   logic circuits operatively coupled to said third system of memory to initialize said DFA in

4   accordance with current state information corresponding to the input channel associated with

5   said data packet.

1   42.   (Original): The classifier of claim 40 wherein some of said states have

2   one or more associated instructions, the classifier further including an arithmetic logic unit

3   operatively coupled to said system of logic circuits and configured to execute said instructions.

1   43.   (Original): The classifier of claim 42 further including at least one register

2   operatively coupled to said arithmetic logic unit, said arithmetic logic unit further configured to

3   store data into said register in response to a save instruction.

1   44.   (Original): The classifier of claim 42 further including skip logic

2   operatively coupled to said logic component and configured to skip over an amount of data in

3   response a skip instruction.

1   45.   (Original): The classifier of claim 40 wherein said stream of data is a

2   stream of bytes.

1   46.   (Original): The classifier of claim 40 wherein said data packets vary from

2   one packet to another.

1   47.   (Previously presented): A network packet classifier comprising:

2   means for receiving an incoming network packet;

3   means for processing said network packet in accordance using a language

4   definition in accordance with a formal language processing technique including classifying said

5   network packet by matching the pattern of constituent data of said network packet against plural

6   regular expressions, each regular expression having a corresponding class identifier; and

7   means for outputting a class identifier of the regular expression which matches

8   said network packet.

1   48.   (Original): The classifier of claim 47 wherein said means for classifying

2   includes a memory component configured with data to represent a deterministic finite automaton

3   (DFA).

1   49.   (Original): The classifier of claim 48 wherein said means for outputting

2   includes a second memory component configured with said class identifiers.

1   50.   (Original): The classifier of claim 47 wherein said regular expressions

2   include arithmetic specifiers and said means for classifying includes an arithmetic logic unit

3   configured to perform operations in accordance with said arithmetic specifiers.

1   51.   (Previously presented): A network packet classifier comprising:

2   a dual-ported memory component;

3   first classification logic operatively coupled to a first port of said dual-ported

4   memory component and having a first input for receiving a data stream; and

5   second classification logic operatively coupled to a second port of said dual-

6   ported memory component and having a second input for receiving a data stream,

7   said memory component configured to contain a deterministic finite automaton

8   (DFA) representative of a language definition and comprising plural states,

9   said DFA representing plural regular expressions for matching data packets,

Appl. No. 09/538,132
Amdt. sent November 29, 2005
Reply to Office Action of June 3, 2005

PATENT

10   said first and second classification logic each configured to process an associated

11   data stream using said language definition according to a formal language processing technique

12   including a step to scan said associated data stream using said DFA to identify data packets

13   contained therein and to classify identified data packets.

1   52.   (Original): The classifier of claim 51 wherein said data packets are

2   characterized in being variable in length.

1   53.   (Original): The classifier of claim 51 wherein said regular expressions

2   include arithmetic and logic operators.

1   54.   (Original): The classifier of claim 51 wherein said regular expressions

2   include a skip operator.